Cairo University
Systems and Biomedical Engineering
Clinical Decision Support Systems

Task 2 Report

Submitted to:
Instructor/Dr. Eman Ayman.
TA/Eng. Abdelrahman.

Presented by: Team Neutron

Ahmed El Sarta
Mahmoud Yasser
Sohaila Mahmoud
Abdelrahman Yasser
Nourhan Sayed

1. Introduction on hypothyroidism

Your thyroid gland controls the metabolism of your body. To stimulate your thyroid, your pituitary gland releases a hormone known as thyroid-stimulating hormone (TSH). Your thyroid then releases two hormones, T3 and T4. These hormones control your metabolism.

In hypothyroidism, your thyroid doesn't produce enough of these hormones.

There are three types of hypothyroidism: primary, secondary, and tertiary.

2. Problem definition

Diagnosing a patient with the 3 levels of hypothyroid, or determining if they are not sick.

The dataset includes 29 features to help predict 4 output classes: negative (not sick), primary, secondary or tertiary hypothyroidism.

The 29 features include:

- Age
- Sex
- Whether a patient is on thyroxine hormone tablets
- Whether a patient is on antithyroid medication
- Whether the patient is sick
- Pregnancy
- Whether a patient had thyroid surgery
- If a patient had I-131 radiotherapy (treatment for hyperthyroidism and thyroid cancer)
- Whether a patient has Hypopituitarism (low supply of pituitary hormone)
- Whether a patient has tumor
- Value of TSH
- Value of T3 hormone
- Value of TT4 hormone
- Value of FTI hormone

3. Data cleaning, visualization, and analysis

- Dataset needed cleaning before being used in the model, as all features' datatypes were objects and multiple features had several unrecorded values. So, after several steps of cleaning and removing outlier points it was all converted to floats and ready to be used in the model.

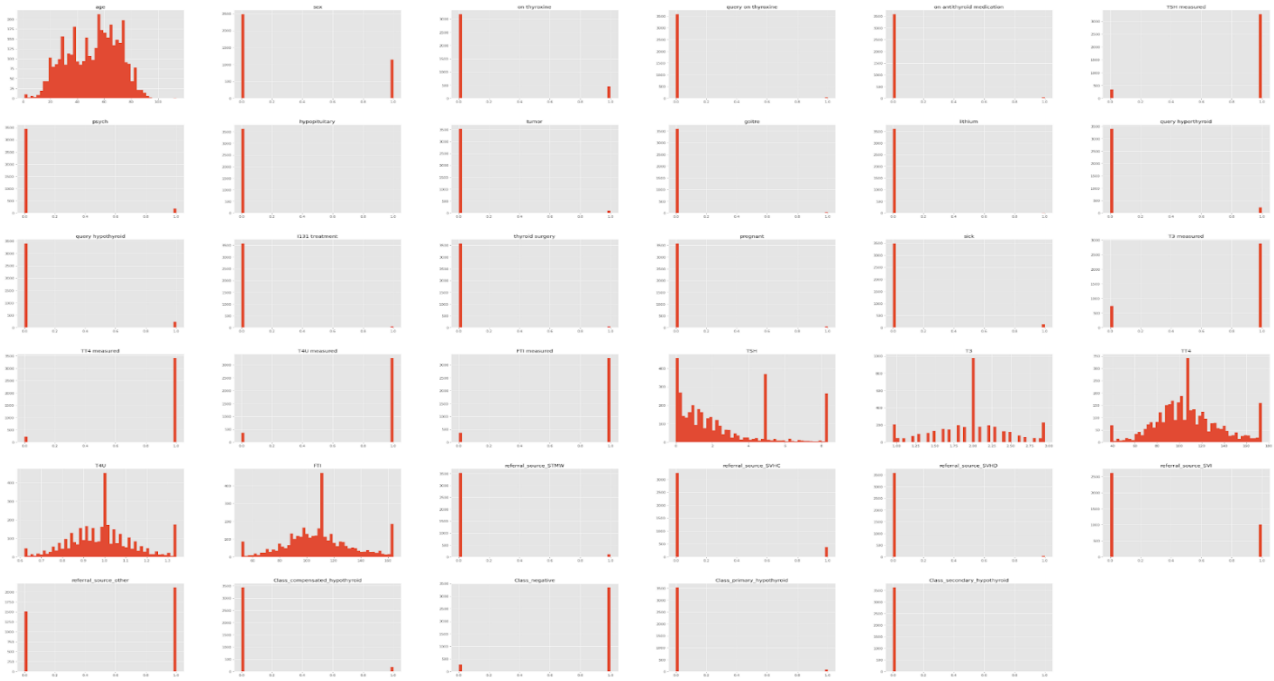
```
df.info()
```

```
Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3772 entries, 0 to 3771
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        3772 non-null   object
 1   sex                        3772 non-null   object
 2   'on thyroxine'             3772 non-null   object
 3   'query on thyroxine'       3772 non-null   object
 4   'on antithyroid medication' 3772 non-null  object
 5   sick                       3772 non-null   object
 6   pregnant                   3772 non-null   object
 7   'thyroid surgery'          3772 non-null   object
 8   'I131 treatment'           3772 non-null   object
 9   'query hypothyroid'        3772 non-null   object
 10  'query hyperthyroid'       3772 non-null   object
 11  lithium                    3772 non-null   object
 12  goitre                     3772 non-null   object
 13  tumor                      3772 non-null   object
 14  hypopituitary              3772 non-null   object
 15  psych                      3772 non-null   object
 16  'TSH measured'             3772 non-null   object
 17  TSH                        3772 non-null   object
 18  'T3 measured'              3772 non-null   object
 19  T3                         3772 non-null   object
...
 28  'referral source'          3772 non-null   object
 29  Class                      3772 non-null   object
dtypes: object(30)
memory usage: 884.2+ KB
```

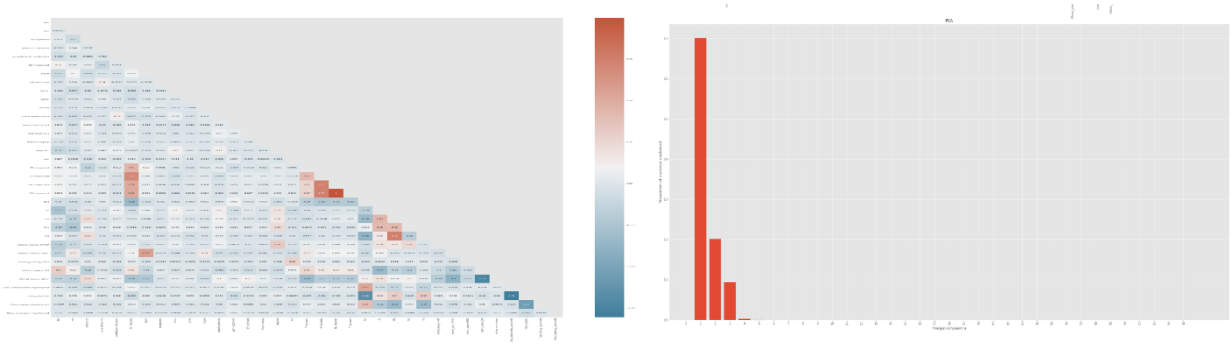
```
df_cleaned = df_cleaned.dropna()
df_cleaned.info()
```

```
Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3622 entries, 0 to 3771
Data columns (total 35 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        3622 non-null   float64
 1   sex                        3622 non-null   float64
 2   on thyroxine               3622 non-null   float64
 3   query on thyroxine         3622 non-null   float64
 4   on antithyroid medication  3622 non-null   float64
 5   TSH measured               3622 non-null   float64
 6   psych                      3622 non-null   float64
 7   hypopituitary              3622 non-null   float64
 8   tumor                      3622 non-null   float64
 9   goitre                     3622 non-null   float64
 10  lithium                    3622 non-null   float64
 11  query hyperthyroid         3622 non-null   float64
 12  query hypothyroid          3622 non-null   float64
 13  I131 treatment             3622 non-null   float64
 14  thyroid surgery            3622 non-null   float64
 15  pregnant                   3622 non-null   float64
 16  sick                       3622 non-null   float64
 17  T3 measured                3622 non-null   float64
 18  TT4 measured               3622 non-null   float64
 19  T4U measured               3622 non-null   float64
...
 33  Class_primary_hypothyroid  3622 non-null   float64
 34  Class_secondary_hypothyroid 3622 non-null  float64
dtypes: float64(35)
memory usage: 1018.7 KB
```

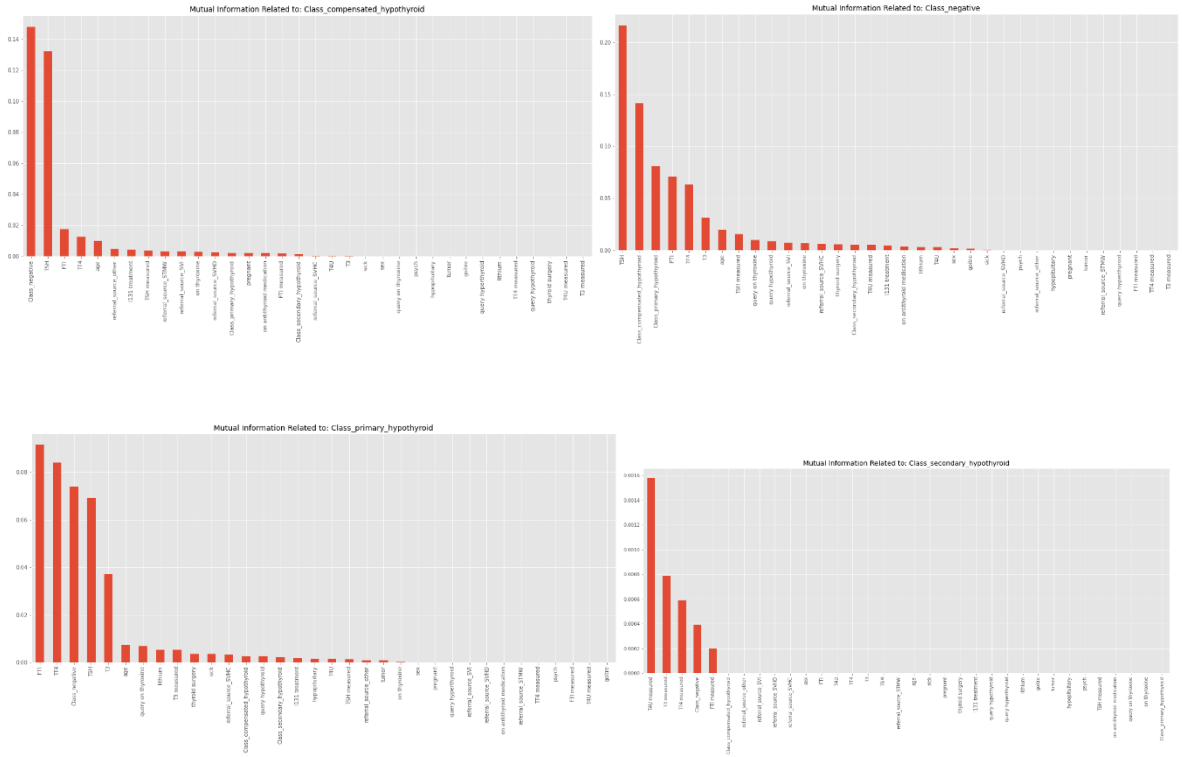
- To visualize the data, for each feature histogram was calculated, and from the figure, it can be observed that many features are normally distributed such as age, t3, tt4, T4U, and FTI, while the binary featured are badly distributed i.e. have imbalanced records, which could lead to overfitting of the model.



- After visualization, analyzation of the data were done using two methods correlation method and Principle Component Analysis (PCA).



- Then next step was using mutual information algorithm for each output class to deduce which features are the best related for the corresponding class.



4. Data Modelling

After the data visualization and analysis techniques, we will try different approaches for modelling.

4.1 The first approach

Using all the features in every model, even the ones with a low correlation to the output. Then we build a model for every output class and compute the accuracy, confusion matrix, and the cross validation score.

The first model, for the "negative" class:
We observe that the output is
Model Accuracy = 1.0
Confusion Matrix = [[88 0] [0 999]]
Average Cross Validation Score = 0.9991712675605545

The second model, for the "compensated" class:
We observe that the output is
Model Accuracy = 1.0
Confusion Matrix = [[1024 0] [0 63]]
Average Cross Validation Score = 0.9997233748271093

The third model, for the "primary" class:
We observe that the output is

Model Accuracy = 1.0
Confusion Matrix = [[1063 0] [0 24]]
Average Cross Validation Score = 0.9988957854668907

The fourth model, for the "secondary" class:
We observe that the output is

Model Accuracy = 0.9990800367985281
Confusion Matrix = [[1086 0] [1 0]]
Average Cross Validation Score = 0.9994478927334453

So, we observe that, although the accuracy is very high with these results (which is almost perfect), the model seems to be overfitting on the training data, which will likely produce bad results on future testing data. So, we move on to try a different approach.

After we visualized the data, it became apparent that some features don't directly affect the output and we could just ignore these features, which lead us to:

4.2 The second approach

Selecting 3 features which highly affect the output and building 4 models based on them.

The first model, for the "negative" class:
We observe that the output is
Model Accuracy = 1.0
Confusion Matrix = [[88 0] [0 999]]
Average Cross Validation Score = 0.9991712675605545

The second model, for the "compensated" class:
We observe that the output is
Model Accuracy = 1.0
Confusion Matrix = [[1024 0] [0 63]]
Average Cross Validation Score = 0.9997233748271093

The third model, for the "primary" class:
We observe that the output is
Model Accuracy = 1.0
Confusion Matrix = [[1063 0] [0 24]]
Average Cross Validation Score = 0.9988957854668907

The fourth model, for the "secondary" class:
We observe that the output is
Model Accuracy = 0.9990800367985281
Confusion Matrix = [[1086 0] [1 0]]
Average Cross Validation Score = 0.9994478927334453

So, we observe that, while the accuracy decreased a little bit, the overfitting also seems to have decreased a little, which makes sense since we dropped some features. But is this the best model we could reach?

4.3 Third approach

Which is to assume 2 classes, whether the patient has hypothyroidism or not. (I.e. we combined the 3 types of hypothyroidism into one class)

Using all features:

```
Model Accuracy = 0.9898804047838087
 F1-score = 0.9897450880335242
 Confusion Matrix = [[996   3]
 [  8  80]]

 Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       999
           1       0.96      0.91      0.94        88

    accuracy                           0.99      1087
   macro avg       0.98      0.95      0.97      1087
weighted avg       0.99      0.99      0.99      1087

 Avg Cross Validation Score = 0.9867494256026885
```

Using only two features, the FTI, TSH:

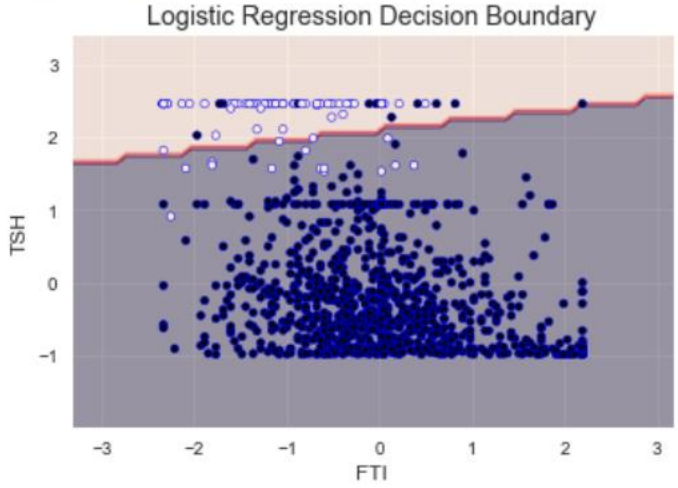
```
Model Accuracy = 0.9733210671573137
 F1-score = 0.9732515147046514
 Confusion Matrix = [[985  14]
 [ 15  73]]

 Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       999
           1       0.84      0.83      0.83        88

    accuracy                           0.97      1087
   macro avg       0.91      0.91      0.91      1087
weighted avg       0.97      0.97      0.97      1087

 Avg Cross Validation Score = 0.9710149400454946
```



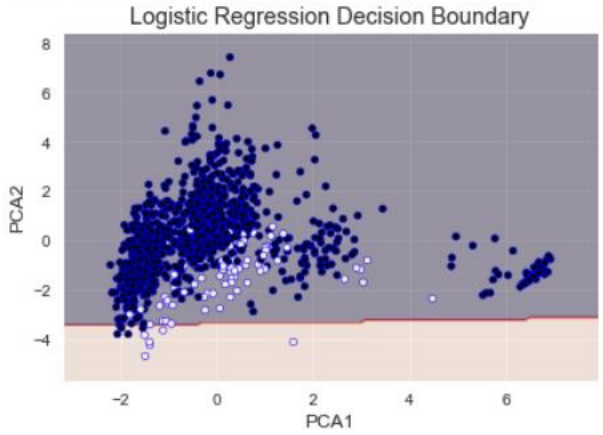
Using PCA features:

```
Model Accuracy = 0.920883164673413
 F1-score = 0.8908841924015619
 Confusion Matrix = [[995   4]
 [ 82   6]]
 Classification Report =
              precision    recall  f1-score   support

           0       0.92      1.00      0.96       999
           1       0.60      0.07      0.12        88

    accuracy                           0.92      1087
   macro avg       0.76      0.53      0.54      1087
weighted avg       0.90      0.92      0.89      1087

Avg Cross Validation Score = 0.9867494256026885
```



We notice that from our 3 scenarios:

- Best Scenario, Can not be plotted => train the Logistic Regression Model with all the features.
- Acceptable Scenario, Can be plotted => train with the best 2 features using the Mutual Information Algorithm.
- Worst Scenario, Can be plotted => train with the PCA features map to the worst metrics as the distribution of the data is *Non-Linear*

5. Final Result

After trying different model approaches, the approach that gave best results and accuracy was using Logistic Regression Model with all the features in case of combination of classes into two main classes.

6. Problems

One of the main problems that faced us was that the dataset needed much of cleaning and work to be able to use it in the model, and for future work it would be preferred to do more cleaning so results becomes more acceptable.